University of Tehran
College of Engineering
School of Electrical and Computer Engineering

Digital Communications

Dr.Rabiei

Computer Assignment 1

Soroush Mesforush Mashhad

SN:810198472

Farvardin 01

Abstract

In this assignment, we start with designing a function which calculates $\mathcal{G}_k$ as discussed in class, it accepts the transition states matrix and k as its inputs.

In the next section we proceed to design a function which accepts a chain of symbols and k as its inputs and computes the average length of the Huffman code if we code the symbols in k-bit groups.

In the next section we proceed to calculate the entropy for the given source analytically, then we create 10 million symbols, and plot its average length and $\mathcal{G}_k$ for the given values for k, we then go on to plot the coding efficiency along with its entropy with respect to k.

In the next part we utilize our previous functions to calculate the average length of the Huffman code and $\mathcal{G}_k$ for the following sources: $\mathcal{X}, \mathcal{X}^2, \mathcal{X}^3$ in which $\mathcal{X}$ is a memoryless source with (0.7,0.29,0.01) as its probabilities. At the end of this part we plot the graphs of the previous section for this scenario.

In the end we shall perform an analysis and comparison between the two previous sections.

1 Problem 10

In this part we shall implement the following equation.

$$\mathcal{G}_k = \frac{\mathcal{H}(S_0) + k\mathcal{H}(S_1|S_0)}{k}$$

In the code when we want to calculate $\mathcal{P}$ from the equation $\mathcal{P}\Phi^T = \mathcal{P}$ we pay attention to the fact that $\mathcal{P}_1 + \mathcal{P}_2 + \mathcal{P}_3 + \mathcal{P}_4 = 1$.

After calculation $\mathcal{P}$ we proceed to calculate $H(S_0)$ with the following formula.

$$\mathcal{H}(S_0) = -\sum_{i=0}^{N} p_i log_2(pi)$$

From here we proceed to calculate $\mathcal{H}(S_1|S_0)$ utilizing the following formula.

$$\mathcal{H}(S_0|S_1) = -\sum_{i=0}^{N} Pr\left[S_0 = S_i\right] \sum_{j=0}^{N} Pr\left[S_1 = S_j|S_0 = S_i\right] log_2(Pr\left[S_1 = S_j|S_0 = S_i\right])$$

Then in the end we plug our results in the following formula.

$$\mathcal{G}_k = \frac{\mathcal{H}(S_0) + k\mathcal{H}(S_1|S_0)}{k}$$

2 Problem 11

In this part we want to design a function to calculate the average length of a Huffman code, first we must check if the chain is divisible by k, if not we remove the extra symbols accordingly. By doing so we modify the chain so it shall be divisible by k.

Next we perform a series of events which all lead to calculating the probability of the k-bit symbol groups.

In the end by using the $huffmandict$ function we calculate the average length.

3 Problem 12

3.1 Analytical Solution

3.1.1 Method 1

In this method we shall use the following formula:

$$\mathcal{G}_k = \frac{\mathcal{H}(S_0) + k\mathcal{H}(S_1|S_0)}{k}, \quad \mathcal{H}(\mathcal{X}) = \lim_{k\to\infty} \mathcal{G}_k = \mathcal{H}(S_1|S_0)$$

We calculate the transition state matrix as follows.

$$\Phi = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.8 & 0.2 \end{bmatrix}$$

Now we have

$$\mathcal{H}(S_0|S_1) = -\sum_{i=0}^{N} Pr\left[S_0 = S_i\right] \sum_{j=0}^{N} Pr\left[S_1 = S_j|S_0 = S_i\right] log_2(Pr\left[S_1 = S_j|S_0 = S_i\right])$$

$$\longrightarrow \mathcal{H}(S_0|S_1) = 0.8930, \quad \mathcal{H}(X) = 0.8930$$

3.1.2 Method 2

In this part we shall calculate the total entropy with the help of the state entropies and the $\mathcal{P}$ vector.

$$\mathcal{H}_i = -\sum_{j=1}^{n} p_{ij} log_2(p_{ij}) \longrightarrow \mathcal{H}_1 = -2 \times 0.5 log_2(0.5) = 1$$

$$\mathcal{H}_2 = -0.2 log_2(0.2) - 0.8 log_2(0.8) = 0.7219, \quad \mathcal{P} = \begin{bmatrix} 0.6154 \\ 0.3846 \end{bmatrix}$$

$$\mathcal{H}(\mathcal{X}) = \sum_{i=1}^{n} \mathcal{P}_i \mathcal{H}_i = 0.6154 + 0.3846 \times 0.7219 = 0.8930$$

As expected the two methods yield the same result.

3.2 Simulation

In this part we create 10 million symbols and satisfy the requirements of the problem with our designed functions from the previous problems.

We know the following:

$$\eta_k = \frac{\mathcal{H}(\mathcal{X})}{\hat{\mathcal{H}}_k}, \quad \hat{\mathcal{H}}_k = \frac{\bar{n}}{k}, \quad \bar{n} : Average\ Length$$

We can prove that when k moves towards infinity then we have $\eta_k = 1$, so we expect $\hat{\mathcal{H}}_k$ to converge to $\mathcal{H}(\mathcal{X})$ therefore to show this I have included $\hat{\mathcal{H}}_k$ in my plots.

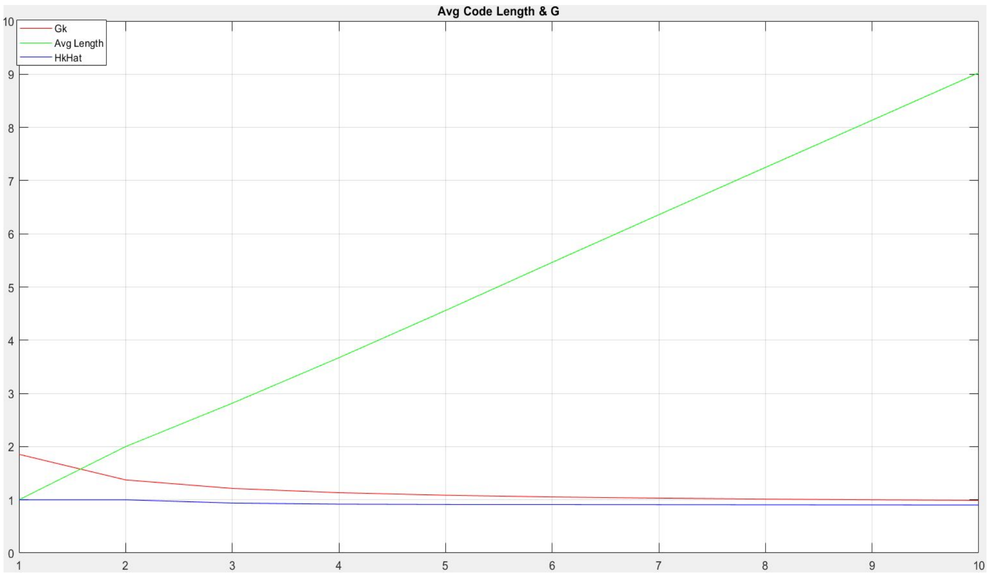

Figure 1: Average code length and $\mathcal{G}_k$

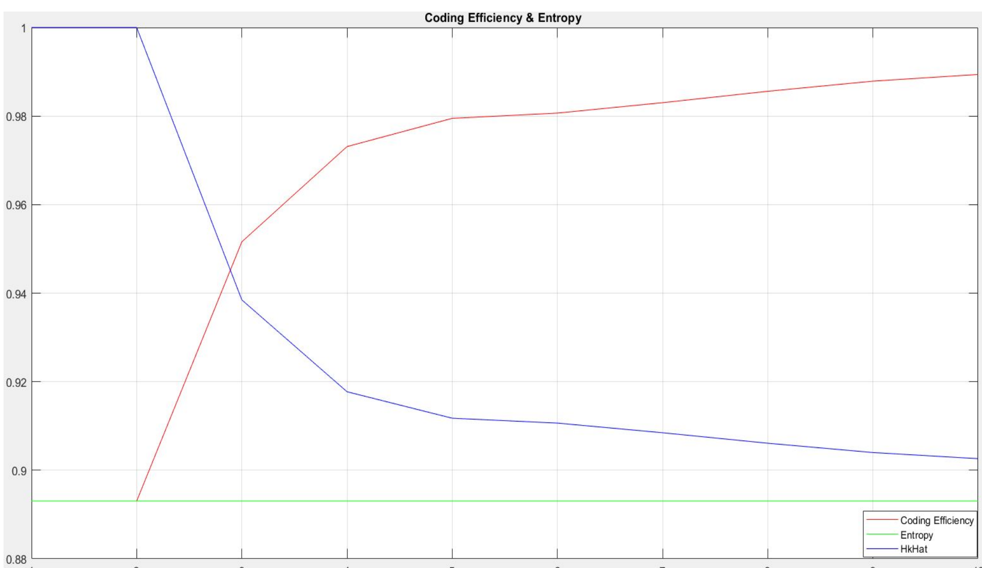

Figure 2: Coding Efficiency and Entropy

It can be easily deduced that by increasing k redundancy is also reduced, the reason I have for this assumption is the almost linear behavior of the average length when k increases, we can also see that by increasing k, $\mathcal{G}_k$ approaches the entropy.

4 Problem 13

In this part we have a memoryless source the probability of its symbols are (0.7, 0.29,0.01), they can have any name for example (1,2,3).

4.1 Analytical calculation of entropy

Due to the fact that the source is memoryless we can calculate its entropy as follows.

$$\mathcal{H}(\mathcal{X}) = -\sum_{i=1}^{n} p_i log_2(pi) = -0.01log_2(0.01) - 0.29log_2(0.29) - 0.7log_2(0.7) = 0.9445$$

we also know that:

$$\mathcal{H}(\mathcal{X}^2) = 2\mathcal{H}(\mathcal{X}) = 1.889, \quad \mathcal{H}(\mathcal{X}^3) = 3\mathcal{H}(\mathcal{X}) = 2.8335$$

The above fact can be proved due to the fact of the source being memoryless and

$$\mathcal{G}_k = \frac{\mathcal{H}(s_0) + k\mathcal{H}(s_0)}{k}, \quad \mathcal{H}(\mathcal{X}) = \lim_{k\to\infty} \mathcal{G}_k$$

4.2 Simulation

We obtain the following graphs with MATLAB.

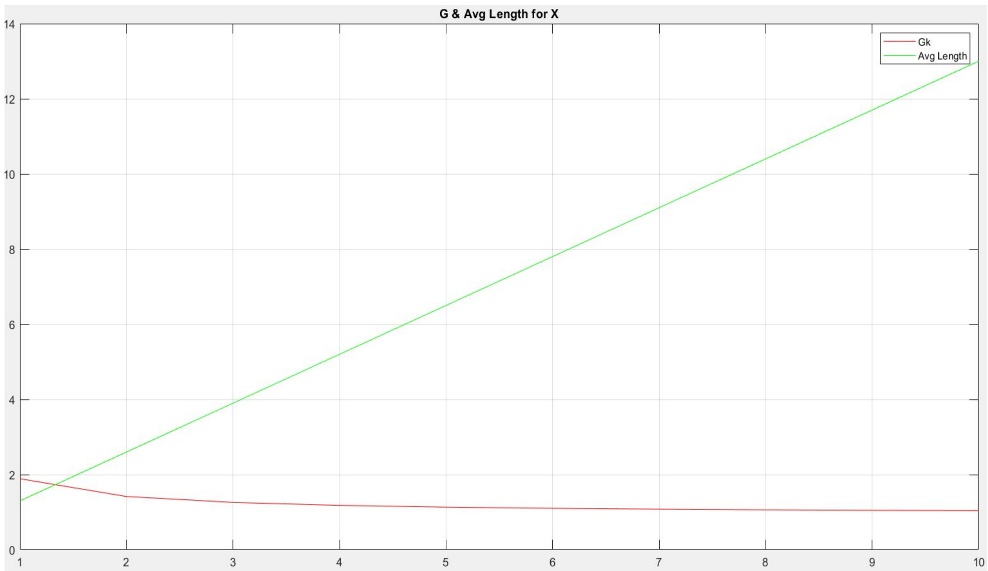

Figure 3: Average code length and $\mathcal{G}_k$ for $\mathcal{X}$

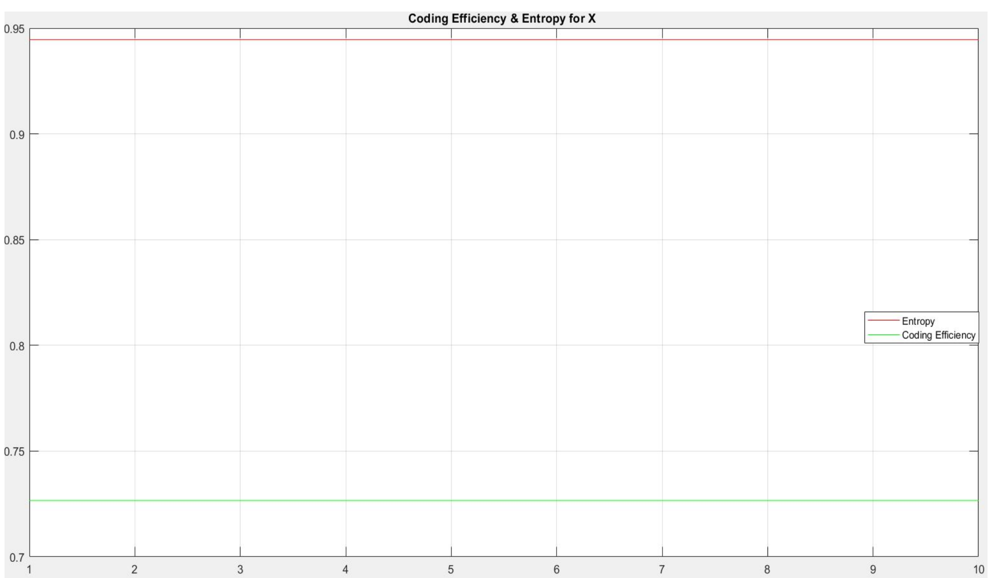

Figure 4: Coding Efficiency and Entropy for $\mathcal{X}$

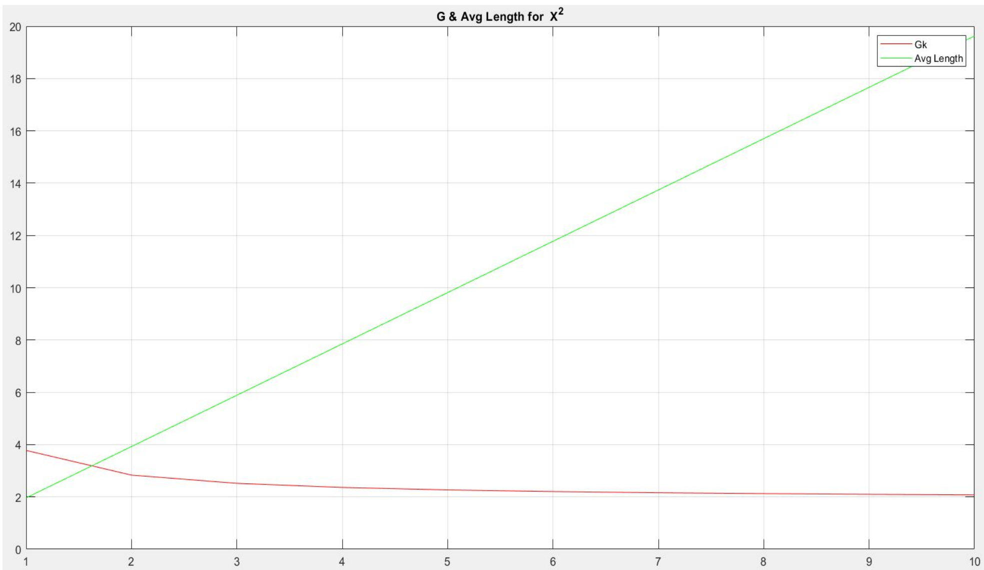

Figure 5: Average code length and $\mathcal{G}_k$ for $\mathcal{X}^2$

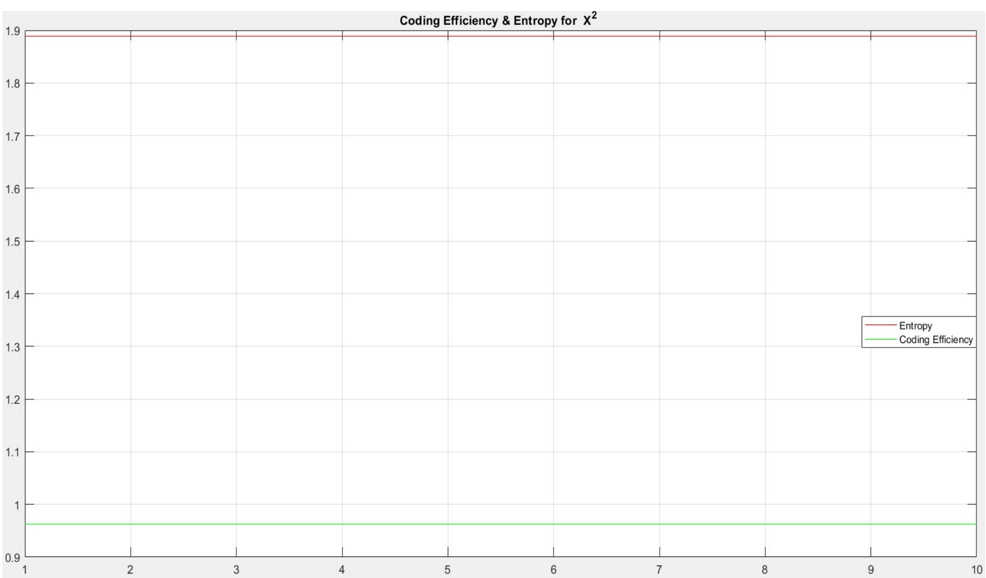

Figure 6: Coding Efficiency and Entropy for $\mathcal{X}^2$

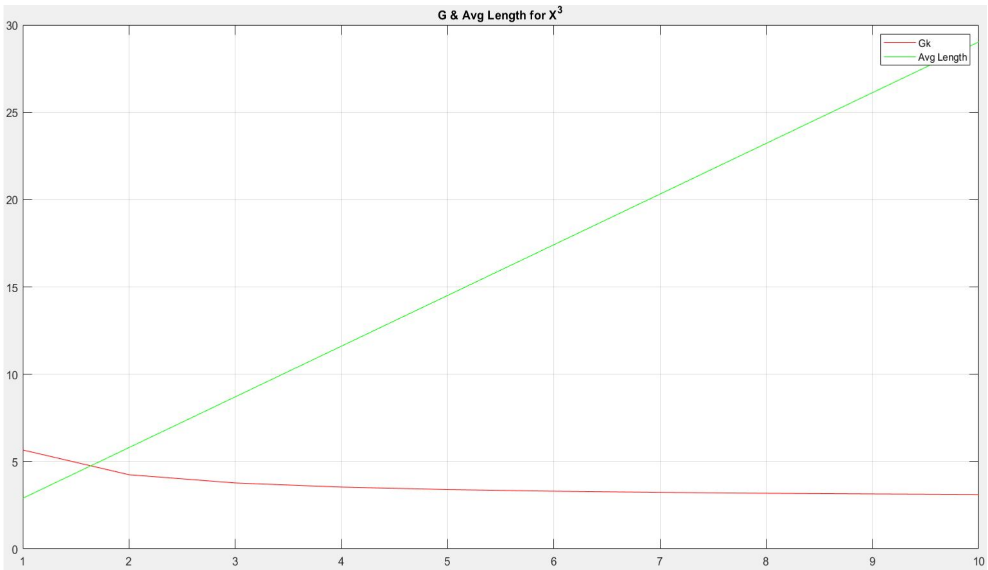

Figure 7: Average code length and $\mathcal{G}_k$ for $\mathcal{X}^3$

Figure 8: Coding Efficiency and Entropy for $\mathcal{X}^3$

As we can see, the average length of the code has a completely linear behavior due to it being memoryless and having zero redundancy, also the coding efficiency shall obviously remain a constant which is clearly shown in the graphs.

4.3 Coding Efficiency

4.3.1 $\mathcal{X}$

$$\eta_k = \frac{k \times \mathcal{H}(\mathcal{X})}{\bar{n}} = \frac{0.9445}{1.3} = 72.6538\%$$

4.3.2 $\mathcal{X}^2$

$$\eta_k = \frac{k \times \mathcal{H}(\mathcal{X})}{\bar{n}} = \frac{1.889}{1.9627} = 96.2449\%$$

4.3.3 $\mathcal{X}^3$

$$\eta_k = \frac{k \times \mathcal{H}(\mathcal{X})}{\bar{n}} = \frac{2.8335}{2.9025} = 97.6227\%$$

The following results can be easily verified by checking the MATLAB graphs and it is also a logical result, we expect the coding efficiency for $\mathcal{X}^3$ to be larger than $\mathcal{X}^2$ and $\mathcal{X}^2$ larger than $\mathcal{X}$ which has occured.

5 Problem 14

Here we compare the memoryless source with the Markov source which is not memoryless.

- Markov sources are affected by the value of k but memoryless sources are not, there is no redundancy in memoryless sources but this is not the case in Markov sources and the redundancy decreases in these sources with the growth of k.

- The coding efficiency for Markov sources differ with the amount of k but is a constant value in memoryless sources.

- The coding efficiency of Markov sources approach 1 with the increase of k but this is not the case in memoryless sources.

- The coding efficiency of memoryless sources can be improved by merging and combining them together.

- Memoryless sources have a higher entropy than Markov sources which is perfectly understandable because the symbols are more ambiguous in this case because there is no information for them when they are generated.